Optimal Task Placement with QoS Constraints in Geo-distributed Data Centers using DVFS

Lin Gu, Deze Zeng, Member, IEEE, Ahmed Barnawi, Member, IEEE, Song Guo, Senior Member, IEEE, and Ivan Stojmenovic, Fellow, IEEE

Abstract—With the rising demands on cloud services, the electricity consumption has been increasing drastically as the main operational expenditure (OPEX) to data center providers. The geographical heterogeneity of electricity prices motivates us to study the task placement problem over geo-distributed data centers. We exploit the dynamic frequency scaling technique and formulate an optimization problem that minimizes OPEX while guaranteeing the quality-of-service, i.e., the expected response time of tasks. Furthermore, an optimal solution is discovered for this formulated problem. The experimental results show that our proposal achieves much higher cost-efficiency than the traditional resizing scheme, i.e., by activating/deactivating certain servers in data centers.

Index Terms—data center, cost minimization, request mapping, dynamic voltage frequency scaling

1 INTRODUCTION

The advantages of cloud computing are increasingly attracting individuals and organizations to move their data and services from local to Internet data centers for reliability, security and expenditure benefits. Many enterprises, such as Amazon, Google, Microsoft, have released various cloud computing services to users. It is a clear and inevitable trend that cloud service is becoming a promising and pervasive service to replace traditional local service.

Cloud services deeply rely on the computation, communication and storage resources in data centers. To the service providers, as shown in [1], electricity cost has become the dominant operational expenditure (OPEX), even surpassing the capital expenditure (CAPEX), i.e., the cost on hardware, due to the extremely high energy consumption of large-scale data centers. Lowering electricity cost has therefore become a major concern.

Modern large-scale data centers are usually deployed in a geographically distributed manner. For example, Google owns and operates many data centers across a wide range of locations, e.g., Council Bluffs, Iowa, Pryor Creek, Oklahoma, Berkeley County, South Carolina, etc., to keep their services available 24 hours a day, 7 days a week [2]. An important fact is that the electricity prices usually vary by time and location, i.e., temporal and spatial heterogeneity. As shown in Fig. 1, the electricity prices fluctuate with time in both Houston and Mountain View, and differ among all three places at any time in a day, e.g., 48.14 and 38.52 for Mountain View and Houston at 13:00, respectively.

Motivated by such fact, considerable effort has been devoted to exploring the data center’s geographical distribution nature and the electricity price heterogeneity to lower the electricity cost [3]–[6].

A promising and representative way is to adjust the number of active servers in different data centers to adapt to the user requests, generally known as data center resizing [3], [7]. An intuitive way is to activate more servers in data center with low electricity price. Then, the following problem should be carefully studied: how to distribute the workload or user requests among all available data centers to achieve low electricity cost while guaranteeing the quality-of-service (QoS), e.g., the expected response time of requests. For example, although greedily powering down the servers with high electricity prices and placing requests to the ones with lower electricity prices can decrease the total energy cost, the QoS may not be guaranteed due to the congestion.
that would happen when relying on less servers. Therefore, the data center resizing and request scheduling are usually jointly considered.

On the other hand, dynamic voltage and frequency scaling (DVFS) has been emerging as a compelling technique for energy efficient computing and has been made available to server-class processors [8]. By DVFS, the frequency or operating voltage of a processor can be dynamically adjusted according to the service requirement. For example, when it is required to complete a large number of user requests within a short delay, the supplied voltage can be increased to speed up the processing rate. Recently, it is even suggested that dynamic frequency scaling shall be also applied to the memory and network for the purpose of approaching the vision of energy-proportional computing [9], [10].

Existing data center resizing methods only focus on the control of turning on or off servers. A fine-grained approach, yet well investigated, is to exploit the benefit of DVFS for energy management in data centers. Different from the conventional pure resizing technique where the processing speed is proportional to the number of activated servers, the first critical problem is the nonlinear relationship between the processing speed and the energy consumption. Using DVFS to decrease the energy consumption of individual processor or cluster has been introduced and analysed in many technical papers, but how to leverage the DVFS technique in data centers to lower the electricity cost is still a new challenge. A study in [11] indicates that DVFS may be beneficial to power reduction in data centers without specifying how to explore DVFS for power reduction.

To conquer above weaknesses and explore the benefits of DVFS technology, we study the cost minimization problem for data center via joint optimization of task placement, data center resizing and frequency adjustment in geo-distributed data centers. Specifically, the spatial-temporal heterogeneity in electricity pricing over geo-distributed data centers is also considered. Our main contributions are summarized as follows:

- Instead of solely relying on data center resizing, we specially investigate how to apply DVFS technique to cut the OPEX in a fine-grained manner. To our best knowledge, we are the first to consider the OPEX minimization problem in geo-distributed data centers with joint consideration of task placement, data center resizing and frequency adjustment.
- We formulate the DVFS technique based OPEX minimization problem into a mixed-integer nonlinear programming (MINLP) to answer the following questions: a) how to place tasks onto servers without violating the computation resource and QoS constraints, b) how to resize each data center, c) how to adjust the work frequency of each server, and d) how to explore the spatial-temporal heterogeneity of geo-distributed data centers to achieve the OPEX minimization goal.
- To combat the high computational complexity of solving the MINLP problem, we explore two problem-specific features and simplify the original formulation into an equivalent, but much simplified, one that can also lead to the optimal solutions in polynomial time. Through extensive real-trace driven simulations, the high efficiency of our DVFS-based scheme is validated by showing significant OPEX advantage over existing state-of-the-art data center management mechanisms.

The rest of the paper is organized as follows. Section 2 summaries the related work. Section 3 gives our system model. The electricity cost optimization problem is formulated in Section 4. An optimal algorithm is proposed in Section 5. The theoretical findings are verified by experiments in Section 6. Finally, Section 7 concludes our work.

2 RELATED WORK

A large number of data centers are being operated by cloud service providers such as Google and Amazon. According to [12], a data center may consist of thousands of servers and consume megawatts of power. Millions of dollars on electricity cost have posed a heavy burden on the OPEX to data center providers. Therefore, reducing the electricity cost has received significant attention from both academia and industry, and various technologies and schemes have been proposed to decrease the energy consumption and electricity cost in data centers both by academia and industry [5], [12]–[14].

2.1 Temporal Energy Buffer

The basic idea of temporal energy buffer technique is to store electricity when the price is low and to drain them during high price period. Govindan et al. [14] use uninterrupted power supply (UPS) units which are normally used as a fail-over mechanism to transition to captive power sources upon utility failure, to store energy during off-peak periods of electricity demand and use it during on-peaks periods of higher demand. Based on this model, they present peak reduction algorithms that combine the UPS battery knob with existing throttling based techniques for minimizing data center power costs. Urgaonkar et al. [5] also explore the energy cost reduction by the use of UPS units as energy storage devices to reduce the electricity bill in data centers. Using Lyapunov optimization, they develop an online control algorithm that can optimally exploit these devices to minimize the time average cost without any knowledge of the statistics of the workloads or electricity prices.

2.2 Request Mapping and Data Center Resizing

Request mapping dynamically dispatches the workloads among all available data centers to explore the electricity price heterogeneities. Data center resizing is a mechanism to reduce the total electricity cost by changing the numbers of activated servers in each data center. They
are usually jointly considered to match the workload requirement. Since the geo-distributed data centers are with various electricity prices and processing abilities, distributing more workloads to data centers with lower price and higher processing abilities will reduce the total electricity cost, but a balance shall also be kept to guarantee QoS.

Gao et al. [15] propose the optimal workload control and balancing by taking account of access latency, carbon footprint, and electricity costs. Liu et al. [16] reduce electricity cost and environmental impact using a holistic approach of workload balancing that integrates renewable supply, dynamic pricing, and cooling supply. Delimitrou et al. [17] present Paragon, a heterogeneity-and interference-aware online DC scheduler, which is derived from robust analytical methods, instead of by profiling each application. Fan et al. [13] study power provisioning strategies on how much computing equipment can be safely and efficiently hosted within a given power budget. Rao et al. [3] investigate how to reduce electricity cost by routing user requests to geo-distributed data centers with accordingly updated sizes that match the requests. Recently, Liu et al. [4] re-examine the same problem by taking network delay into consideration.

### 2.3 Power Gating

Any running circuits will cause both static and dynamic power consumption. The former only relates to the circuit on/off status, and the latter changes according to the workload. Power gating [18] is a promising technique to reduce the static power consumption by deactivating unused circuit units (e.g., processing unit). The potential of power gating in server energy saving has been proved in [19] and has also attracted much interest in the research on data center energy management. Leverich et al. [20] argue that power gating is an efficient technique for power management of data center workloads. Bilgir et al. [21] also verify such viewpoint by showing that 35% energy saving can be achieved by appropriate core count selection and mapping in data centers. For online data-intensive services in data centers, Meisner et al. [22] propose a data center energy saving mechanism by applying power gating technique.

### 2.4 DVFS

DVFS technology has been deployed into different hardware architectures to reduce the dynamic energy consumption. Howard et al. [23] describe a multi-core processor that integrates 48 cores, 4 DDR3 memory channels, and a voltage regulator controller in a 64 2D-mesh network-on-chip architecture which can operate at different frequencies. Rountree et al. [24] provide a first look at this technology in high performance computing environment and detail both the opportunities and potential pitfalls of using this technique to control processor power.

Hanumaiah et al. [25] address the problem of determining the feasible speeds and voltages of multi-core processors with hard real-time and temperature constraints. Beyond processors, recently Fu et al. [26] propose an energy-aware architecture for routers, which could trade system performance for energy savings while traffic is low by scaling frequencies of its inner components. Etinski et al. [27] present a model that gives an upper bound on performance loss due to frequency scaling using the application parallel efficiency and study how cluster power consumption characteristics together with application sensitivity to frequency scaling determine the energy effectiveness of the DVFS technique. Li et al. [28] consider scheduling frame-based tasks on DVFS-enabled heterogeneous multiprocessor platform with the goal of achieving minimal overall energy consumption.

Previous work on geo-distributed data center management only focuses on data center resizing and workload distribution. The conventional resizing technique may cause a high electricity cost because the electricity consumption of servers is fixed regardless of the amount of workload. When a server is activated but not fully used, i.e., the workload is lower than the full processing capability, it will cause significant waste due to the cube relationship between power consumption and processing speed. For example, the power consumption of a half-used server can decrease by 87.5% if the DVFS technique is adopted instead of using fixed frequency.

### 3 System Model

In this section, we introduce the data center model, the DVFS technique and the workload model. For the conveniences of the readers, the major notations used in this paper are listed in Table 1.

### 3.1 Geo-distributed Data Centers

We consider a general structure with $I$ geo-distributed data centers and $J$ front portals at different regions. Fig. 2 gives an example where there are 4 data centers and 3 front portals, i.e., $I = 4$ and $J = 3$. The front portal in each region is in charge of collecting and distributing the regional user requests. A data center

---

**Table 1**

<table>
<thead>
<tr>
<th>Notation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$I$</td>
<td>total number of data centers</td>
</tr>
<tr>
<td>$S_i$</td>
<td>total number of servers in data center $i$</td>
</tr>
<tr>
<td>$N_i$</td>
<td>total number of activated servers in data center $i$</td>
</tr>
<tr>
<td>$X_{is}$</td>
<td>requests allocated to server $s$ in data center $i$</td>
</tr>
<tr>
<td>$P_{Is}$</td>
<td>requests allocated to data center $i$ from front portal $j$</td>
</tr>
<tr>
<td>$f_{Is}$</td>
<td>frequency of server $s$ in data center $i$</td>
</tr>
<tr>
<td>$J$</td>
<td>total number of front portals</td>
</tr>
<tr>
<td>$P_{fr}$</td>
<td>the electricity power cost of data center $i$ from front portal $j$</td>
</tr>
<tr>
<td>$D$</td>
<td>the maximum expected response time</td>
</tr>
<tr>
<td>$P_{data}$</td>
<td>static power consumption</td>
</tr>
</tbody>
</table>
usually has a large number of servers. Let \( S_i \) represent the number in data center \( i \) (\( i \in [1, I] \)). We consider a multi-electricity market, where the electricity prices exhibit location diversities. The electricity price for data center \( i \) is denoted as \( P_{R_i} \).

### 3.2 Dynamic Voltage and Frequency Scaling

DVFS is a computer architecture technique that allows dynamically adjusting the operating frequency of a microprocessor via regulating the supplied voltage. Higher voltage supply indicates faster frequency and larger power consumption meanwhile. The power consumption can be written as a function of the supply voltage \( V \) and the operating frequency \( f \), i.e.,

\[
P = C_0 \cdot V^2 \cdot f + P_{\text{static}},
\]

where \( C_0 \) is the effective capacitance and \( P_{\text{static}} \) is the static power consumption [8], [11], [29], [30], which is mainly incurred by current leakage and is independent of \( f \). Existing work uncovers that \( P_{\text{static}} \) can be regarded as a constant, which varies from 10% to 60% due to the different architectures and technologies, where \( f_{\text{max}} \) is the maximum frequency that can be supported [13], [30], [31]. Generally, the supply voltage has a power relationship with the operating frequency, i.e., \( V \propto f^\alpha \), where \( \beta \) is approximately within 1.0 to 1.3 for current technology [29], [30]. Letting \( \alpha = 2 \cdot \beta \), the supply voltage can be rewritten as

\[
P = C \cdot f^{\alpha+1} + P_{\text{static}}.
\]

### 3.3 Workload Model

Without loss of generality, we assume the arrival of user requests at each front portal as a Poisson process. Such assumption has been proved by real-trace studies (e.g., [32], [33]) and also widely adopted in the literature (e.g., [13], [15]–[17]). We denote the user request arrival rate at front portal \( j \) (\( j \in [1, J] \)) as \( \lambda_j \). A user request can be distributed to any data center, or more specifically, any server. The distribution of a request from the front portal to a server in a data center is generally known as request-mapping in the literature [34]. The front portal distributes a request with a predetermined probability to a server such that the user request arrival in a server can be also regarded as a Poisson process. Let us denote the average request arrival rate at server \( s \) in data center \( i \) from the front portal \( j \) as \( \lambda^{js}_{ji} \), \( s \in [1, S] \).

In modern data centers, management servers are responsible for task placement to the processing servers, each of which has a local buffer to maintain the allocated tasks [35], [36]. For a given server frequency, it is commonly recognized that the processing time satisfies exponential distribution and therefore the dynamics of a server’s task queue can be described using an M/M/1 queuing model [37]. The service rate in a DVFS-enabled server is proportional to its frequency, i.e., \( \mu_s = \gamma \cdot f^*_s \), where \( f^*_s \) is the processing frequency of server \( s \) in data center \( i \) and \( \gamma \) is the scaling factor.

### 4 PROBLEM FORMULATION

#### 4.1 Power Consumption Constraints

Let \( P_i^s \) be the power consumption of server \( s \) in data center \( i \) as an example. A server’s power consumption is first determined by its activation status, e.g., \( P_i^s = 0 \) when it is deactivated. When it gets activated, \( P_i^s \) is related to both the operation frequency \( f^*_s \) and usage, i.e.,

\[
P_i^s = C \cdot (f^*_s)^{\alpha+1} \cdot p_i^s \cdot f^*_s + P_{\text{static}},
\]

where \( p_i^s \) is the probability that server \( s \) in data center \( i \) is busy. According to the M/M/1 queuing theory, \( p_i^s \) can be expressed as:

\[
p_i^s = \frac{\lambda_i}{\gamma \cdot f^*_i},
\]

where

\[
\lambda_i = \sum_{j=1}^{J} \lambda_{ji}^j
\]

is the total requests dispatched from all \( J \) front portals.

Using a binary variable \( \chi^s_i \in \{0, 1\} \) to denote whether a server is activated \( \chi^s_i = 1 \) or not \( \chi^s_i = 0 \), \( P_i^s \) can be written in a uniform way as

\[
P_i^s = \chi^s_i \cdot (C \cdot (f^*_s)^{\alpha} \cdot \frac{\lambda_i}{\gamma} + P_{\text{static}}).
\]

The relationship between \( \chi^s_i \) and \( f^*_s \) can be represented by

\[
\frac{f^*_s}{f_{\text{max}}} \leq \chi^s_i \leq f^*_i.
\]

The correctness of (7) can be validated as: when \( f^*_s = 0 \), \( 0 \leq \chi^s_i \leq 0 \) indicates that \( \chi^s_i = 0 \) while \( \chi^s_i \equiv 1 \) if \( f^*_s > 0 \) since \( 0 \leq f^*_s \leq f_{\text{max}} \).
4.2 Workload Balance Constraints

The user requests arrive at front-portal \( j \) with a rate \( \lambda_j \) and are then distributed to a server for processing. The requests allocated to data center \( i \) is the sum of all requests processed in all servers as \( \sum_{j=1}^{J} \lambda_j \). The total requests to be processed in all servers shall be equal to the ones placed at all front portals, i.e.,

\[
\sum_{j=1}^{J} \lambda_j = \sum_{i=1}^{I} \sum_{s=1}^{S_i} \lambda_s^i.
\] (8)

4.3 QoS Constraints

As we mentioned in Section 3, the request processing procedure of server \( s \) in data center \( i \) can be viewed as a queuing system with the mean arrival and service rate of \( \lambda_s^i \) and \( \mu_s^i = \gamma \cdot f_s^i \). Under the existence of steady state, the expected delay \( d_s^i \) at each server can be written as:

\[
d_s^i = \frac{1}{\gamma \cdot f_s^i - \lambda_s^i}.
\] (9)

To satisfy the QoS requirement, the expected delay \( d_s^i \) at any activated server (i.e., \( \chi_s^i = 1 \)) shall not exceed the delay constraint \( D \). This leads to

\[
f_s^i \geq \frac{\chi_s^i}{D \cdot \gamma} + \frac{\lambda_s^i}{\gamma}, \forall i \in [1, I], s \in [1, S_i],
\] (10)

because \( d_s^i \leq D \) is achieved when \( \chi_s^i = 1 \) and (9) becomes \( \lambda_s^i = 0 \) when \( \chi_s^i = 0 \).

4.4 A MINLP Formulation

Recall that all servers in data center \( i \) shall share the electricity price \( P_r^i \). The total electricity cost can then be calculated by summing up the cost on each server across all the geo-distributed data centers, i.e.,

\[
C_{\text{total}} = \sum_{i=1}^{I} \sum_{s=1}^{S_i} \chi_s^i \cdot (C \cdot (f_s^i)^\alpha + \frac{\lambda_s^i}{\gamma} + P_{\text{static}}) \cdot P_r^i.
\] (11)

Our objective of minimizing the electricity cost can be realized by the best settings of \( \chi_s^i, f_s^i \) and \( \lambda_s^i \). Summarizing all constraints discussed above, the electricity price minimization problem can be formulated as a mixed-integer nonlinear programming (MINLP) problem as:

\[
\begin{align*}
\text{MINLP:} \\
\min_{\chi_s^i, f_s^i, \lambda_s^i} & : J \sum_{i=1}^{I} \sum_{s=1}^{S_i} \chi_s^i \cdot (C \cdot (f_s^i)^\alpha + \frac{\lambda_s^i}{\gamma} + P_{\text{static}}) \cdot P_r^i, \\
\text{s.t.} : & \sum_{j=1}^{J} \lambda_j = \sum_{i=1}^{I} \sum_{s=1}^{S_i} \lambda_s^i, \\
& f_s^i \geq \frac{\lambda_s^i}{\gamma} + \frac{\chi_s^i}{D}, \forall i \in [1, I], s \in [1, S_i], \\
& \frac{f_s^i}{f_{\text{max}}} \leq \chi_s^i \leq f_s^i, \forall i \in [1, I], s \in [1, S_i], \\
& 0 \leq f_s^i \leq f_{\text{max}}, \forall i \in [1, I], s \in [1, S_i].
\end{align*}
\]

5 Algorithm Design

Different from the pure resizing technique, the processing capability of a data center capable of DVFS is not only determined by the number of activated servers but also the processing frequencies of these servers. As a result, there are two methods, i.e., activating more servers and increasing the processing frequencies of activated servers, to increase a data center’s processing ability. An interesting question is on which decision shall be made to provide enough processing ability to guarantee the QoS of tasks allocated to a data center. As shown in (11), whenever a server is activated, a basic static power consumption is incurred, regardless of its processing frequency. On the other hand, if we increase the activated servers’ processing frequencies, the processing ability will increase linearly but the power consumption will increase with a cube relationship at the same time.

The above discussion is only about the electricity cost minimization of one data center under a given workload. To data center providers, it is desired to minimize the total cost of all data centers. From this point of view, the workloads allocated to data centers shall be considered systematically with the consideration of electricity price diversity. Intuitively, it is preferable that more workloads shall be distributed to the data center with low electricity price. However, aggressive distribution to such a data center may require activating all servers and making them operate at a high frequency to satisfy the QoS of all tasks. The electricity cost may be sharply increased due the cubic relationship between power consumption and processing frequency. This may even counteract the advantage of exploring data center with low electricity price.

To tackle the above issues, we first introduce two theorems as the foundations of our algorithm design.

**Theorem 1:** For a given request demand, the minimal power consumption in a data center is achieved when the user requests are uniformly distributed among all activated servers.

**Proof:** From (6), we observe that the power consumption function is an increasing function of server frequency \( f_s^i \). For electricity cost minimization, it is reasonable to have

\[
f_s^i = \frac{\lambda_s^i}{\gamma} + \frac{1}{\gamma \cdot D}.
\]

without violating the QoS constraint. Thus, the minimum power consumption \( P_s^i \) can be rewritten as a function of \( \lambda_s^i \), i.e.,

\[
P_s^i(\lambda_s^i) = C \cdot (\frac{\lambda_s^i}{\gamma} + \frac{1}{\gamma \cdot D})^\alpha \cdot \frac{\lambda_s^i}{\gamma} + P_{\text{static}}.
\]

Denoting the number of activated servers in data center \( i \) as \( N_i \), the total power consumption can be calculated as \( \sum_{s=1}^{N_i} P_s^i(\lambda_s^i) \). As \( P_s^i(\lambda_s^i) \) is strictly convex in range \((0, \infty)\), we can apply Jensen’s inequality and get

\[
\sum_{s=1}^{N_i} P_s^i(\lambda_s^i) \geq P_s^i\left(\frac{\sum_{s=1}^{N_i} \lambda_s^i}{N_i}\right) \cdot N_i,
\] (12)
where equality holds if and only if $\lambda^*_1 = \lambda^*_2 = \cdots = \lambda^*_N$. That is to say, the minimum power consumption of a data center can be achieved by uniformly distributing all the requests to all activated servers.

Let $\nu_i = \sum_{x=1}^{S_x} \lambda^*_x$ denote the total requests distributed to data center $i$. According to Theorem 1, the optimal power consumption at data center $i$ for given $N_i$ can be rewritten as

$$ P_i = N_i \cdot \left( C \cdot \left( \frac{\nu_i}{\gamma \cdot N_i} + \frac{1}{\gamma \cdot D} \right)^\alpha \cdot \frac{\nu_i}{\gamma \cdot N_i} + P_{\text{static}} \right) \cdot Pr_i, $$

(13)

Accordingly, we can reformulate problem MINLP to:

**MINLP2:**

$$ \min_{\nu_i, N_i} : \sum_{i=1}^{I} N_i \cdot \left( C \cdot \left( \frac{\nu_i}{\gamma \cdot N_i} + \frac{1}{\gamma \cdot D} \right)^\alpha \cdot \frac{\nu_i}{\gamma \cdot N_i} + P_{\text{static}} \right) \cdot Pr_i, $$

s.t.: $\sum_{j=1}^{J} \lambda^*_j = \sum_{i=1}^{I} \nu_i$,

$$ \frac{\nu_i}{\gamma \cdot N_i} + \frac{1}{\gamma \cdot D} \leq f_{\text{max}}, \forall i \in [1, I], $$

$$ N_i \leq S_i, \nu_i \geq 0, \forall i \in [1, I]. $$

Without loss of generality, let $Pr_1 \leq Pr_2 \cdots \leq Pr_I$ be the electricity price list in a decreasing order of all data centers. The corresponding solution $\{N_1, \ldots, N_I\}$ is named server activation configuration (SAC). While problem formulation given in MINLP2 is still in a MINLP form, the following theorem about the property of SAC allows it to be solved in a much reduced complexity.

**Theorem 2:** In the optimal SAC, there exists $i, 1 \leq i \leq I$, such that for any $1 \leq n \leq I$,

$$ N_n = S_n, \quad n < i, $$

$$ N_n \leq S_n, \quad n = i, $$

$$ N_n = 0, \quad n > i. $$

(14)

**Proof:** Suppose there exists an optimal SAC $S^*$ that does not satisfy the format, i.e., $\exists N_x \neq S_x$, $N_y \neq 0, 1 \leq x < y \leq I$, like $\{S_1 = 1, S_2 = 2, \ldots, S_I = 1\}$ shown in Fig. 3(a). According to Theorem 1, the corresponding request arrival rates to all servers shall be $\frac{\nu_i}{\gamma \cdot N_i}, \forall N_i \neq 0, i \in [1, I]$, resulting in total electricity cost $C'_{\text{total}}$.

Consider another solution $S'$ obtained by migrating all the requests from a server in data center $y$ to a deactivated one in data center $x$ and denote the total cost after the migration as $C''_{\text{total}}$. Note that such migration will not introduce any violation to the constraints in MINLP2. For example, after migrating all the requests from a server in data center $I$ to data center 1 in Fig. 3(a), we come to a new SAC $\{S'_1, S'_2, \ldots, S'_I\}$ shown in Fig. 3(b). The difference $C''_{\text{total}} - C'_{\text{total}}$ can be calculated as:

$$ (C \cdot \left( \frac{\nu_y}{\gamma \cdot N_y} + \frac{1}{\gamma \cdot D} \right)^\alpha \cdot \frac{\nu_y}{\gamma \cdot N_y} + P_{\text{static}} \right) \cdot (Pr_x - Pr_y), $$

(15)

by which we can derive $C''_{\text{total}} < C'_{\text{total}}$ since $Pr_x < Pr_y$. We reach a contradiction that $S'$ is not the optimal solution.

We call the SAC format specified in Theorem 2 as optimal format henceforth. Obviously, combinatorial number of possible optimal format is in a polynomial order of $S_1, 1 \leq i \leq I$. Furthermore, in MINLP2, we notice that the objective function is convex and all the constraints are affine when the SAC is fixed. This implies that the Karush-Kuhn-Tucker (KKT) conditions are necessary and sufficient for the globally optimal solution. Therefore, for any given SAC, MINLP2 can be rewritten to a Lagrangian Dual Problem and solved using KKT conditions in polynomial time [38]. We define the Lagrangian:

$$ L(a, b, \nu) = f(\nu) + \sum_{i=1}^{I} a_i g_i(\nu) + bh(\nu), $$

(16)

$$ g_i(\nu) = \frac{\nu_i}{\gamma \cdot N_i} + \frac{1}{\gamma \cdot D} - f_{\text{max}}, $$

(17)

$$ h(\nu) = \sum_{j=1}^{J} \lambda^*_j - \sum_{i=1}^{I} \nu_i, $$

(18)

where $\nu = [\nu_1, \nu_2, \ldots, \nu_I]$, $a$ and $b$ are Lagrange multipliers vectors, and $f(\nu)$ is the objective functions in MINLP2. Then we could obtain the optimal solution by solving the following KKT conditions:

$$ \frac{\partial L(a, b, \nu)}{\partial \nu_i} = 0, \forall i \in [1, I], $$

(19)

$$ a_i \cdot g_i(\nu) = 0, \forall i \in [1, I], $$

$$ g_i(\nu) \leq 0, h(\nu) = 0, \forall i \in [1, I], $$

$$ a_i \geq 0, \forall i \in [1, I]. $$

Thus, for a given SAC, we can get its corresponding minimal cost by solving (19) as it is equivalent to solving MINLP2. By iterating all possible SACs in optimal format, we are able to obtain the optimal minimal cost. However, this is still time-consuming for large-scale data centers with a mass number of servers since there are $\sum_{i=1}^{I} S_i$ optimal formats. Other than searching all possible formats in a brute-force way, we design a two-phase algorithm where we first obtain an initial SAC with the maximum possible number of activated servers and then iteratively update the SAC by reducing the number of activated servers to find out the optimal solution.
by solving MINLP2. The algorithm is summarized in Algorithm 1.

The first phase between lines 1 and 21 is electricity price independent but only focuses on the power efficiency in all data centers. Due to the existence of static power consumption $P_{\text{static}}$, there is a choice between activating more servers or increasing the operation frequency to match a given request demand, as shown in (13). From another point of view, there exists a minimum request rate $\nu_{\text{min}}$ that requires all servers activated in data center $i$ to achieve the highest power efficiency. Treating $P_i$ in (13) as a function of $\nu_i$ and $N_i$, i.e., $P_i(N_i, \nu_i)$, $\nu_{\text{min}}$ can be obtained as $\nu_{\text{min}} = \arg\min_{\nu_i}(P_i(S_i, \nu) \leq P_i(S_i - 1, \nu))$. This value is helpful for us to identify the SAC with maximum possible activated servers.

If the total request rate exceeds $\sum_{i=1}^I \nu_{\text{min}}$, all data centers are supposed to be fully activated as all their power efficiencies can be ensured (see line 19). Accordingly, the maximum number of data centers with activated servers (i.e., $N_i \neq 0$) $I_{\text{max}}$ is also registered as $I$, as shown in line 17. Otherwise, we try to activate the data centers iteratively from the one with the lowest electricity price until all user requests can get satisfied, as shown between lines 8 and 16. Whenever the remained user requests $\nu^r$ is larger than or equal to $\nu_{\text{min}}$, we fully activate data center $i$, i.e., $N_i = S_i$ (line 13). Otherwise, if $\nu^r < \nu_{\text{min}}$, we find out the maximum number of servers that shall be turned on to guarantee the data center power efficiency, as shown in line 10. After that, since all requests have been satisfied, there is no need to activate more servers. We terminate the loop for finding initial SAC and set $I_{\text{max}}$ as $i$ in lines 11 and 17, respectively.

After the first phase, we obtain the SAC with the maximum servers that shall be activated as well as its corresponding optimal cost $C_{\text{total}}$ (line 22). Involving more servers than this SAC will degrade the power efficiency and therefore does not help with electricity cost reduction. However, it is possible to power down one or more servers since the initial SAC is obtained when each data center is allocated the minimum request rate for full activation. Promoting the request rates on some cheaper data centers may reduce the number of expensive servers and potentially lower the total cost. To always obey the optimal SAC format, we iteratively try to deactivate servers in the most expensive activated data center starting from data center $I_{\text{max}}$ until no gain can be obtained, as shown between lines 25 and 41. Then, we apply binary-search concept to find the number of servers that shall be activated in the most expensive activated data center so as to obtain the optimal solution. For each new SAC', its corresponding optimal cost $C'_{\text{total}}$ is obtained by solving MINLP2. If a comparatively lower cost is achieved, we register it as the temporarily optimal cost (line 31) and try to deactivate more expensive servers (line 32). However, once the new cost $C'_{\text{total}}$ is larger than the temporarily optimal cost, it implies that we have already overly deactivated the servers. No further deactivation shall be taken and instead we shall try to recover some overly deactivated servers thereafter. Note that we first set a flag “over_deactivation” as false, indicating that no over-deactivation happens, before we start searching the optimal SAC in line 24. Therefore, in this case, we set “over_deactivation” as true (line 34) and the whole algorithm shall be ended after the searching in current data center ends (lines 38-40).

**Remark:** Algorithm 1 can obtain the optimal solution of MINLP with a computational complexity of $O(n \log n)$. Note that the dominant complexity of Algorithm 1 is on the second phase, where the most data centers are iteratively deactivated between lines 25 and 41. For a data center $i$ with $S_i$ activated servers in the initial SAC, $\log_2 S_i$ iterations are required. In the worst case, all $I$ data centers shall be examined and therefore totally $\sum_{i=1}^I \log_2 S_i = O(n \log n)$ iterations are needed. Also note that, under a given SAC in the optimal format, we always apply KKT conditions to solve MINLP2 with convex objective and affine constraints to get the globally optimal solution in polynomial time.

![Graph](image_url)

**Fig. 4.** On the Effect of $P_{\text{static}}$
Algorithm 1 Optimal Server Activation and Task Placement

Input: $I, J, \lambda, Pr_i, S_i, i = [1, I], j = [1, J]$  
Output: SAC, $C_{\text{total}}$, $\nu_i, i = [1, I]$  
1: Sort data centers in ascending order of $Pr_i, i = [1, I]$  
2: for $i = 1, 2, \cdots I$ do  
3: $\nu_i^{\text{min}} = \arg \min_{\nu_i} (P_i(S_i, \nu) \leq P_i(S_i - 1, \nu))$  
4: end for  
5: if $\sum_{j=1}^{J} \lambda_j < \sum_{i=1}^{I} \nu_i^{\text{min}}$ then  
6: SAC $\leftarrow \{N_i = 0, \forall i \in [1, I]\}$  
7: $\nu' = \sum_{j=1}^{J} \lambda_j$  
8: for $i = 1$ to $I$ do  
9: if $\nu' < \nu_i^{\text{min}}$ then  
10: $N_i = \arg \min_N P(N, \nu') \leq P(N - 1, \nu')$  
11: break  
12: else if $\nu' \geq \nu_i^{\text{min}}$ then  
13: $N_i = S_i$  
14: $\nu' = \nu' - \nu_i^{\text{min}}$  
15: end if  
16: end for  
17: $I_{\text{max}} = i$  
18: else  
19: SAC $\leftarrow \{N_i = S_i, i = [1, I]\}$  
20: $I_{\text{max}} = I$  
21: end if  
22: $\{\nu_i, i = [1, I]\}, C_{\text{total}} \leftarrow$ solve MINLP2 under SAC  
23: SAC$' = \text{SAC}, \nu'_i = \nu_i, \forall i = [1, I]$  
24: over_deactivation=false  
25: for $i = I_{\text{max}} \text{ to } 1$ do  
26: $N_e = 0, N_e = N_i$  
27: while $N_e \leq N_e$ do  
28: update SAC$'$ by setting $N_i = \lceil \frac{N_e + N_e}{2} \rceil$  
29: $\{\nu'_i, i = [1, I]\}, C'_{\text{total}} \leftarrow$ solve MINLP2 under SAC$'$  
30: if $C'_{\text{total}} < C_{\text{total}}$ then  
31: $C_{\text{total}} = C'_{\text{total}}$  
32: $N_e = N_e - 1$  
33: else  
34: over_deactivation=true  
35: $N_e = N_e + 1$  
36: end if  
37: end while  
38: if over_deactivation==true then  
39: return  
40: end if  
41: end for

6 PERFORMANCE EVALUATION

In this section, we present our performance evaluation results on our optimal algorithm by comparing with the conventional pure resizing method discussed in [3] and power gating [20]. A geo-distributed data center simulator following the model described in Section 3 is implemented using Matlab. The two algorithms are named as “Resizing” and “PG”, respectively. Same to [3], real data center information and electricity prices are applied to our experiments. Three Google data centers and four Google front portals in US are considered, i.e., $I = 3$ and $J = 4$. The electricity prices $Pr_i$ in the data center locations, i.e., Mountain View, Houston and Atlanta, are shown in Fig. 1. We consider three data centers with 15000, 30000 and 10000 double-core servers. A core’s service rate is linearly proportional to its operation frequency $f$. We define the maximum service rate of one core as 1 unit per second when $f = f_{\text{max}} = 1$GHz. The request rates at the four portals are 30000, 15000, 15000 and 20000 units per second. Regarding the QoS requirement, we set the maximum expected delay $D$ as 0.01 second. For computation tractability, voltage-frequency relationship factor is set as 1, i.e., $\alpha = 2$ and the peak dynamic power consumption is normalized as 1, i.e., $C f_{\text{max}} = 1$. Unless explicitly specified, we select the electricity prices at 9:00 as 42.93, 20.27 and 55.30 for the three data centers, respectively.

6.1 Performance Comparison under Different Settings

As we have known, the value of $P_{\text{static}}$ has a deep influence on the data center management and hence the total electricity cost. Therefore, we first evaluate the total electricity power costs under different values of $P_{\text{static}} \in [0.1, 0.6]$. Specially, in order to validate the optimality of our algorithm, we also consider a small-scale data center with $S_i = \{15, 30, 10\}$ and $\lambda_i = \{30, 15, 15, 20\}$. MINLP is solved using a Matlab function fminconset for MINLP problem [39]. The results are shown in Fig. 4. At first, from Fig. 4(a), we can see that the results of Algorithm 1 tightly match with the ones obtained by solving MINLP. This validates the optimality of our algorithm. From both figures, we can see that the total cost shows as an increasing function of the static power consumption $P_{\text{static}}$, in any algorithm. For any $P_{\text{static}}$, we can always observe the substantial advantage of “DVFS” over “Resizing” and “PG”, especially when $P_{\text{static}}$ is small.

An interesting observation is that such advantage diminishes with the increasing of $P_{\text{static}}$. For example, in Fig. 4(b) “DVFS” saves 30% electricity cost compared with “Resizing” and “PG” when $P_{\text{static}} = 0.1$. However when $P_{\text{static}}$ increases to 0.6, such advantage drops to 10%. This is because to cut the data center power consumption with QoS guarantee, there are two factors: static and dynamic power consumption. For example, deactivating more servers (cores) will decrease the static power consumption while lowering the servers’ frequencies can cut the dynamic power consumption. When $P_{\text{static}}$ is low, e.g., 0.1, the dynamic power consumption takes most part of the total power consumption. In this case, lowering the servers’ frequencies will exert great positive effects on cutting the total power consumption as well as the total cost. Therefore, our proposal “D-VFS” shows significant advantage over the other two
algorithms. However as $P_{\text{static}}$ becomes high, e.g., 0.6, increasing activated servers' frequencies becomes more power efficient than activating a server (core) according to (13). The servers' operation frequencies may be set to a high value, even approaching $f_{\text{max}}$. In this case, less space is left for frequency optimization and therefore the advantage degrades. Fortunately, it is reported that nowadays the maximum $P_{\text{static}}$ is up to 60% of the peak power consumption and researchers are still actively seeking various ways to lower its proportion.

Fig. 5 illustrates the total electricity power cost as the function of total user request rate from 30000 to 110000 and Fig. 6 plots the total electricity power cost under different server numbers in the three data centers, which increase from $\{20000, 50000, 100000\}$ to $\{400000, 700000, 300000\}$. We first notice that the cost shows as an increasing function of the total request rate and a decreasing function of server numbers. Furthermore, the advantage of "Optimal" over "Resizing" can be always observed except when the rate reaches 110000 or the server numbers are $\{20000, 50000, 10000\}$. This is because in this case almost all the servers shall be turned on and run in maximum frequency, i.e., $N_i = S_i$ and $f_i^* = f_{\text{max}}$. No space is left for cost minimization using frequency scaling. Another notable observation is that "DVFS" outperforms "Resizing" and "PG" significantly under all different settings. There are two reasons: 1) When the ratio of the static power consumption to dynamic one is low, adjusting server frequencies plays a more important role than simply activating/deactivating servers or cores. 2) In our scenario, data centers are geo-distributed with various electricity price. Therefore, during optimization it is natural to avoid using high-price servers. There will be less optimization space left for pure on/off algorithms, like "Resizing" and "PG".

6.2 Real Task Trace Driven Cost Analysis

In this section, we evaluate our optimal algorithm during 24 hours in a day using some real trace. Firstly, we construct 3 front portals to generate our service requests using a real trace of 1998 World Cup [40] and their hourly numbers of requests are shown in Fig. 7. All these requests will be processed by 3 data centers in Mountain View, Houston and Atlanta with the hourly electricity power prices shown in Fig. 1. As shown in Fig. 8, the total electricity power cost is significantly reduced in every hour. This further proves that our DVFS-aware data center management scheme can indeed cut the OPEX in a multi-electricity market with price diversities.

Fig. 9, Fig. 10 and Fig. 11 give the detailed values of the request distributions, activated server numbers, and processing frequencies of all 3 data centers, respectively. It can be observed from Fig. 9, the workload in data center 2 is much higher than the other two. This is because data center 2 owns the largest number of servers and is always of the lowest price. Another interesting point is that the workload of data center 1 is always higher the that of data center 3 except the time period from 15:00 to 18:00 due to the larger number of servers and lower electricity price. While observed form Fig. 1, during 14:00 to 18:00 the electricity price of data center 1 exceeds the that of data center 3 therefore the workload of data center 1 decreases while that of data center 3 increases. The similar trend can also observed from Fig. 10 for the same reasons.

From Fig. 9 and Fig. 10, we also observe that the workload distribution and numbers of activated servers correlate to the combination of total workloads from all front portals, hourly price and the numbers of servers in each data center. As the total workloads change from time to time according to Fig. 7, the servers with the lowest price are always activated, while the number of activated servers in high-price data centers will decrease during the low workloads periods, while increase in high workloads periods. For example, during 1:00 to 9:00 almost all servers are deactivated in data center 3, and during 11:00 to 15:00 all servers are activated in all three data centers.

The processing frequencies during 24 hours are given in Fig. 11, showing that frequencies are highly related with hourly electricity prices of data centers. We can
see that data center with the highest price will process in the lowest frequency. For example, at 2:00, the frequency of data center 2 is highest while that of data center 3 is lowest. From Fig. 11, we can observe that the process frequency of the low-price servers is not set to a high value, e.g., 0.51 at 8:00 and 0.61 at 16:00, much lower than the maximum value $f_{\text{max}}$. This is because frequency is in a quadratic relationship of the power consumption. Even in a low-price data center, a small increasing in frequency will result in higher power consumption. Therefore, instead of only exploring the process ability of the low-price servers (“Resizing” and “PG”), “DVFS” not only considers the diversity of the hourly electricity prices, but also tries to balance the electricity cost of each individual data center. Fig. 10 and 11 further detail how “DVFS” can minimize the electricity cost. It can be seen that the servers with the highest electricity price are always deactivated while frequencies of all activated servers are carefully adjusted to minimize the cost.

7 CONCLUSION

The recent increasing demands on cloud computing has made the electricity cost become the dominant OPEX of data centers. To cut the OPEX by reducing the electricity cost, in this paper, we study on a DVFS-aware data center management and request scheduling algorithm by exploring the data center’s geo-distribution and the electricity price diversities.

The optimization problem is first formulated to a MINLP problem. To find an efficient solution, we first prove the existence of optimal server activation config-
uration format and then propose an iterative searching algorithm that can achieve the optimal performance in polynomial time. Through real data based experiments, we show that taking DVFS into fine-grained data center management has substantial advantage over traditional pure resizing based coarse-grained scheme to lower the OPEX on electricity cost.

To further explore the potential of DVFS, our future work includes online requests scheduling and data center management.

ACKNOWLEDGEMENT

This research is supported by the Fundamental Research Funds for the Central Universities, China University of Geosciences (Wuhan) (Grant No. CUG140615). This research of Ivan Stojmenovic is partially supported by NSERC Discovery grant (Canada), and TR32016, Serbian Ministry of Science and Education (associated with Singidunum University, Serbia).

REFERENCES


Lin Gu received her M.S. degrees in computer science from University of Aizu, Fukushima, Japan, in 2011. She is currently pursuing her Phd in University of Aizu, Japan. She received his B.S. degree in computer science and technology from School of Computer Science and Technology, Huazhong University of Science and Technology, China in 2009. She is a student member of IEEE. Her current research interests include cloud computing, vehicular cloud computing, Big Data and Software-defined Networking.

Deze Zeng received his Ph.D. and M.S. degrees in computer science from University of Aizu, Aizu-Wakamatsu, Japan, in 2013 and 2009, respectively. He received his B.S. degree from School of Computer Science and Technology, Huazhong University of Science and Technology, China in 2007. He is currently an associate professor in School of Computer Science, China University of Geosciences (Wuhan), China. His current research interests include: cloud computing, software-defined sensor networks, data center networking, networking protocol design and analysis. He is a member of IEEE.

Ahmed Barnawi is an associate professor at the Faculty of Computing and IT, King Abdulaziz University, Jeddah, Saudi Arabia, where he works since 2007. He received PhD at the University of Bradford, UK in 2006. He was visiting professor at the University of Calgary in 2009. His research areas are cellular and mobile communications, mobile ad hoc and sensor networks, cognitive radio networks and security. He received three strategic research grants and registered two patents in the US. He is IEEE member.

Song Guo (M’02-SM’11) received the PhD degree in computer science from University of Ottawa, Canada. He is currently a Full Professor at School of Computer Science and Engineering, the University of Aizu, Japan. His research interests are mainly in the areas of protocol design and performance analysis for computer and telecommunication networks. He received the Best Paper Awards at ACM IMCOM 2014, IEEE CSE 2011, and IEEE HPCC 2008. Dr. Guo currently serves as Associate Editor of IEEE Transactions on Parallel and Distributed Systems and IEEE Transactions on Emerging Topics in Computing. He is in the editorial boards of ACM/Springer Wireless Networks, Wireless Communications and Mobile Computing, and many others. He has also been in organizing and technical committees of numerous international conferences, including serving as a General Co-Chair of MobiQuitous 2013. Dr. Guo is a senior member of the IEEE and the ACM.

Ivan Stojmenovic was editor-in-chief of IEEE Transactions on Parallel and Distributed Systems (2010-3), is Associate Editor-in-Chief of Tsinghua Journal of Science and Technology, and is founder of four journals (Journal of Multiple Valued Logic and Soft Computing, Ad Hoc & Sensor Wireless Networks, International Journal of Parallel, Emergent and Distributed Systems, Cyber Physical Systems). He is editor of IEEE Transactions on Computers, IEEE Network, IEEE Transactions on Cloud Computing etc. Stojmenovic has top h-index in Canada for mathematics and statistics, and has >16000 citations and h=63. He received five best paper awards. He is Fellow of IEEE, Canadian Academy of Engineering and Academia Europaea. He received Humboldt Research Award in Germany and Royal Society Research Merit Award in UK. He is Tsinghua 1000 Plan Distinguished Professor.